



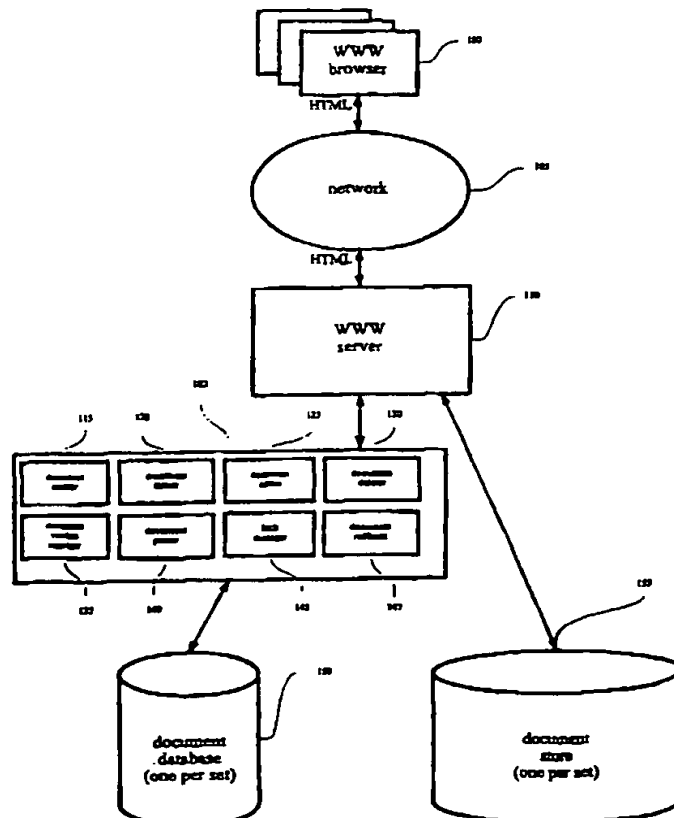
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30	A1	(11) International Publication Number: WO 98/18087 (43) International Publication Date: 30 April 1998 (30.04.98)
(21) International Application Number: PCT/GB97/02887 (22) International Filing Date: 20 October 1997 (20.10.97) (30) Priority Data: 9621763.3 18 October 1996 (18.10.96) GB (71) Applicant (for all designated States except US): BRITISH TELECOMMUNICATIONS PLC [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB). (72) Inventor; and (75) Inventor/Applicant (for US only): BENYON, Paul, Robert [GB/GB]; 27 Chandos Drive, Martlesham, Woodbridge, Suffolk IP12 4TA (GB). (74) Agents: MUSKER, David, Charles et al.; R.G.C. Jenkins & Co., 26 Caxton Street, London SW1H 0RJ (GB).		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: DATA TRANSMISSION AND EDITING

(57) Abstract

Apparatus for use in a file storage and transfer system, the apparatus comprising a web server with a file store for storing at least a first HTML document, and a communications port for sending said first document via the Internet towards a terminal computer running a web browser, at which it may be edited and for receiving an edited file therefrom, arranged to recognise separate sections of said document, in which during stages of said editing, only sections of said document which are to be edited are carried over the Internet.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

DATA TRANSMISSION AND EDITING

This invention relates to apparatus and a method for editing and managing files of data (e.g. representing text documents) over networks of
5 connected computers.

Many computers internationally are connected together through communication links and networks. The development of common communications protocols has enabled diverse such computers to behave in some respects like a single, geographically dispersed network, which has
10 popularly been termed the Internet.

One mechanism by which files on one computer may be accessed and transmitted to another is provided by the use of hypertext links, which are widely used at present, notably in what is popularly referred to as the Worldwide Web (WWW).

15 Many host computer sites are programmed to provide hypertext server programs such as httpd (HyperText Transfer Protocol Daemon), and many network access terminals (e.g. personal computers) are programmed to operate hypertext communications client programs (popularly referred to as "browsers") such as Mosaic (TM) (available from the National Centre for Supercomputing
20 Applications (NCSA) at the University of Illinois; Explorer (TM) (available from Microsoft Inc.); and Netscape Navigator (TM), (available from Netscape). Such client programs are arranged to communicate, via a telecommunications

channel (typically including an end user link such as a local telephone line and a higher capacity link through a network access provider such as Compuserve or the present applicant) to transmit to the host computer running the server a program request for a particular file and, on receipt of the file, are arranged to display the file contents.

Most files are in the form of documents in Hypertext Markup Language (HTML). Such documents comprise ASCII text characters, interspersed with control character strings, in the form of commands which are interpreted by the browser program to control the display of the document. Since the browser program may run on many different types of terminal computer, with different sized displays, different fonts, and different graphics capabilities, the actual format of the document display is determined by the browser together with the user interface of the terminal computer, based on the commands embedded in the hypertext document received from the server.

Many of the commands in hypertext documents simply relate to traditional features of document layout, such as paragraph breaks, line feeds and font size.

Another type of hypertext command comprises the specification of an image file address or file name. This causes the browser program to retrieve the specified image file and display the corresponding image at the point in the document at which the command was placed (i.e. as an "in line image").

However, distinguishing hypertext documents from other documents are the provision of hypertext links or "anchors" within the document. Each link consists of:

1. A defined point in the document at which some displayed matter (e.g. a word or a picture) acts as a "soft button" which may be selected by a user via a user interface comprising, for example, a mouse or other pointing device, or arrow keys, and

2. The address or file name of a further file which, when the user selects the corresponding area of the document as described, will be retrieved by the browser program and displayed in place of the current display.

Many longer hypertext document include internal hyperlinks, to other parts of the same document. Typically, the document is broken into several sections, each commencing with a section heading, and at the start of the document (i.e. the first displayed portion) is an index listing the section headings, or, more generally, the location within the document to which instant access might be required.

Items in the list of headings in the index each comprise a hyperlink, and at the corresponding section heading in the document to each hyperlink is provided a HTML command specifying that it is a label to which a link may point.

Other forms of file transfer mechanism are used across computer networks. For example, for sending files across the Internet, a command protocol referred to as ftp (file transfer protocol) is widely used.

At present, hypertext transfer of documents is widely used as an
5 information access method for retrieving documents; in other words, is widely used for unidirectional information transfer, with client computer terminals operating browser programs requesting the downloading of large volumes of data, but transmitting back only relatively low bandwidth command and communication signals.

10 Some hypertext documents, such as subscription documents or documents for which use is to be charged, contain a form to be completed by the user at the terminal computer. The form typically requests the insertion of the name, address and other bibliographic information concerning the user, and may require the insertion of brief comments, or the selection of technical
15 information (e.g. the model of computer used) from amongst a list of items.

Having completed the form, typically, the user will select a hypertext link button at the end of the form which causes the browser to transmit back the contents of the form to the server via the communications channel.

Aspects of the present invention are concerned with the provision of a
20 document authoring system which allows the text of a document to be developed, by repeated editing over time. Typically, such a document finally

reaches a stage in which it is finalised, at which point it may be "issued" or "released".

Often, documents such as technical specifications may be the product of several authors, all of whom will need access to the document during development. Rather than developing a document on the word processor of a single user, it is therefore known to make the document available over a local area network (LAN) for a number of users to work on it.

Many technical documents, such as product use manuals or system specifications, are subject to periodic changes, which are often minor and affect only a small part of a very large document.

An aim of the present invention is to make a document development system available over a wide area network such as the Internet.

HTML documents at present are typically developed and edited on a terminal computer such as a personal computer, and then uploaded onto a server computer. The HTML documents may be created and edited directly, using a text editor and word processor with ASCII capability.

An alternative is to edit a word processor document to add hypertext commands, using an HTML editor or filter program such as HTML assistance; HoTMetaL available from SoftQuad Inc.; cu_html; Web Author available from Quarterdeck Systems Inc.; or Internet Assistant available from Microsoft Inc. Some such programs can be used with a browser program, so that the result of the edited text can be viewed through the browser program.

A particular problem associated with the use of the Internet (but which could equally be found in other networks) is the effect of network congestion on file transfer speed. Since data transfer across the Internet backbone links is connectionless, the effect of increased traffic is to statistically increase the delay experienced in transferring an entire file for all users. The popularity of some Internet server computers has lead to user frustration at the delay encountered in transferring data, and may deter commercial uses of the Internet.

Were a document authoring system to be provided over the Internet, this could lead to the transfer of large volumes of additional information over the network since, for every (typically minor) change, an entire (typically long) document would be transferred.

This could cause a higher risk of transmission failure during sending a document, as well as greater delays for all users.

In one aspect, therefore, the present invention provides a document communication and editing system, for example for use over the Internet, in which at least the section of a document to be edited is downloaded in an editable format; edited; and then the edited section is transferred back to the server computer. This can therefore significantly reduce the volume of traffic crossing the network. Furthermore, this can significantly reduce the risk of transmission failure since only a section of the document needs to be successfully transmitted.

This may be achieved by providing, at a server computer, that the section to be edited is transferred in an HTML form, which can be interpreted and used by a currently available HTML browser such as Netscape Navigator (TM). Thus, this enables the present invention to be implemented entirely by software at the server computer, requiring no alteration to the software or hardware available at each terminal computer. An embodiment of the invention can therefore use a conventional hypertext browser program at each terminal computer without requiring any special hypertext editor program.

Other aspects and embodiment of the invention are described and claimed hereunder, with advantages that will be apparent from the following description and claims.

The invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1a is a structural block diagram illustrating the hardware present in an embodiment of the invention;

Figure 1b is a functional block diagram illustrating the program elements configuring the hardware of Figure 1a;

Figure 2 (comprising Figures 2a & 2b) is a flow diagram illustrating the general operation of the embodiment of Figure 1, and particularly the server computer thereof;

Figure 3 is a flow diagram illustrating the operation of the document creation process of Figure 2;

Figure 4 is a flow diagram illustrating the operation of the document editing process of Figure 2;

Figure 5 is a flow diagram illustrating the operation of the document issuing process of Figure 2; Figure 6 is a flow diagram illustrating the operation of the document issuing process of Figure 2; and

Figure 7 is a flow diagram illustrating the operation of the document version rollback process of Figure 2.

Referring to Figure 1a and Figure 1b, in one embodiment, the present invention comprises a host computer 20 such as a SUN webserver including a CPU 21; communication port 22 and memory/storage 25 configured to act as a worldwide web server under control of a worldwide web server program 110 loaded and running within the memory 25 of the computer 20.

The computer 20 is connected, via the Internet 105, to a conventional computer terminal 10 within the memory of which is loaded and operating a worldwide web browser program such as Netscape Navigator (TM).

Also provided and running within the computer 10 are a user interface such as a GUI (and, for example, comprising Windows 3.1 (TM) or Windows 95 (TM), available from Microsoft Inc.) and supporting user interface features such as input means 12 comprising a keyboard and pointer device such as a mouse; and user output means 13 such as display on a visual display unit (not shown) or printer (not shown) together with local storage and file handling on a

store such as a hard disk drive 14, all controlled by a CPU 15, and communicating via a port 16 (e.g. a modem) with the Internet 105.

Further provided at the server computer 110 are a document store 155 storing one or more hypertext document made available by the server computer
5 20 via the Internet 105; a document management program 112 via which access to the documents in the document store 155 is performed according to the invention; and a document database 150 storing details of each document for use by the document management program 112.

The document management program 112 comprises a document creator
10 subprogram 115; a document issuer subprogram 120; a document editor subprogram 125; a document deleter subprogram 130; a document version manager subprogram 135; a document parser subprogram 140; a lock manager subprogram 145; and a document rollback subprogram 147.

15 In the present embodiment, the server 110 is a conventional httpd program, and is operable to allow access from a terminal 10 operating under control of a browser program 100 to the documents within the document store 150 exactly in the manner of currently operating server programs. This mode of operation is available for finalised documents, is unidirectional, and does not
20 permit editing of the documents. The skilled reader will be familiar with the operations involved, which will not be described further herein.

The document management program 112 is, in this case, a cgi-bin program resident on the server computer.

Overview of operation of the embodiment

The embodiment permits a user of the terminal computer 10 to create, edit and manage a collection of HTML documents on the server computer 20, and also allows access to other users to permit them to co-author documents.

In greater detail, it allows the user to:

1. create a new document;
2. edit the new document to produce a new version thereof, which is allocated a new version number, whilst maintaining previous versions;
3. transform the current version of a document into a new issue of the document, deleting all working versions back to the last issue (if any), but maintaining previous issues and allocating a new issue number to the current issue;
4. delete a document, including all copies of the document.

To enable these operations, for each document, a document history record is maintained which lists each issue of the document, and all versions (if any) since the last issue, together with their dates of creation.

Additionally, an attempt is made to ensure that only one person is editing the document at a time, by warning subsequent users who attempt simultaneously to access a document that the document is already being edited.

Overview of method of operation

Rather than accessing the document directly, according to the embodiment, documents are accessed via the document manager program 112, which reads the data in the document database 150 for the document for which access is sought, and supplies necessary information to the terminal computer
5 10 via the server program 110.

Where the document is to be edited, the program 102 reads the document from the document store 155, amends the document for editing, and forwards the amended document via the server program 110 to the user terminal browser program 100.

10 On receipt of an uploaded edited version of the document via the server program 110, the document manager program 102 reconverts the document to standard HTML for storage in the document store 155.

Detailed description of operation of the invention

As has been indicated above, the operation of the browser program 100
15 on the user terminal 10 is generally conventional, as is that of the file server program 110.

The operation of the document management program 102 will now be described in greater detail, with reference to Figure 2.

In step 200, the prospective author asks for access to a particular
20 document via the browser program 100 (step 200) by typing the uniform resource locator (URL) address of the document into the URL entry field of the browser program 100. The URL may be typically of the form:

httpd://(host computer)/(name of document management program path)/(path)/(document file name).htm

Within the communication signals from the browser program 100 is an identification of the users terminal computer 10 or, more generally, the user
5 (typically, the Internet address of the terminal computer 10). In step 205, the server program 110 determines, in conventional fashion, whether the identified user is permitted access to the document set identified by the URL. If not, then in step 210, access is refused an the process terminates.

If the user is permitted access to the set, then in step 215 the server
10 program 110 causes the server computer 20 to transmit back a signal asking for the user name and password of the user. On receipt of these, in step 220 the server program 110 determines whether the user name and password are valid; if not, access is refused (step 210) and the process terminates. If the user is identified as a valid user, then (step 225) the document version manager
15 subprogram (135) is called.

The document version manager 135 causes the server computer 20 to read the document database 150, which stores, for each document (or, more accurately, for each set of documents, where a set comprises all the versions and issues of the document) a list of all previous issues of the document and the
20 versions (sub issues) since the last issue, together with the dates on which each was created.

The document version manager 135 generates, from these records, a list of all the documents (document sets) held in the document store 155, together with a hypertext link to an index page to each document (to be discussed below), and a hypertext link labelled "create new document". This HTML page is then transmitted to the user terminal 10 at which it is displayed by the browser program 100.

Where the user selects the "create new document" hyperlink (by positioning the cursor over the corresponding portion of the displayed page and clicking, or pressing the "enter" key), the browser program 100 causes the user terminal computer 10 to transmit back a signal specifying the URL of the document manager program 102 and calling the document creator subprogram (step 230 and 235). The actions performed by the document creator subprogram 115 will be discussed in greater detail hereafter.

The only other valid response which a user can make (if he wishes to continue the interaction with the document management and editing system) is to select (step 240) one of the documents from the document list contained in the displayed page (e.g. by positioning the cursor on the desired document within the list and clicking).

The effect of so doing is to activate the hypertext link to that documents, causing the browser program 100 to cause the terminal computer 10 to transmit a URL consisting of:

(document URL)/index/index.cgi

On receipt of this URL at the server computer 20, the server program 110 passes the URL to the document management software 102. In response (step 245) the document version manager subprogram 135 accesses the document database 150 for the document identified in the URL; reads the historical information concerning the document; and returns to the server program a document history page in HTML format, comprising a series of hypertext links to issues of the document, together with links to versions of the documents since the last issue, dates of creation and identifications of authors/editors for each issue or version.

Also present in the HTML page are hyperlinks entitled:

"view an existing version";
"edit document";
"issue document";
"delete document"; and
"rollback to a previous version".

This HTML document is transmitted by the server program 110 via the network 105 to the terminal computer 10 where it is displayed by the browser program 100.

The user may be content to view the contents for information purposes, in which case no further action is taken by the system.

Alternatively, the user may select one of the buttons displayed which correspond to the hyperlinks embedded in the HTML document index page. In

this case, the browser program 100 causes the terminal computer 10 to transmit a corresponding URL to the server computer 20, which is passed to the document management program 102 by the server program 110.

The "view an existing version" button simply specifies the URL of the document itself in the document store 155. If (step 250) the user has signalled an intention to view an existing version of the document, the document is retrieved from the document store 155 and forwarded to the server 110. The document is transmitted from the server 110 to the user terminal 10 where it is caused to be displayed by the browser program 100 (step 255).

If (step 260), the user has selected the document edit button, the browser program 100 causes the terminal computer 10 to transmit a URL via the Internet 105, which is recognised by the server 110 and passed to the document management software 102, which calls the document editor subprogram 125 (step 265) to be discussed in greater detail hereafter.

If (step 270), the user has selected the document issuer button, the browser program 100 causes the terminal computer 10 to transmit a URL via the Internet 105, which is recognised by the server 110 and passed to the document management software 102, which calls the document issuer subprogram 120 (step 275) to be discussed in greater detail hereafter.

If (step 280), the user has selected the document delete button, the browser program 100 causes the terminal computer 10 to transmit a URL via the Internet 105, which is recognised by the server 110 and passed to the

document management software 102, which calls the document delete subprogram 130 (step 285) to be discussed in greater detail hereafter.

Document Creation (115 Figure 1b; step 235, Figure 2)

Referring to Figure 3, the steps comprised by the document creator subprogram 115 will now be discussed in greater detail.

In a step 300, the document creator subprogram creates a hypertext document including a form which contains a field to allow the insertion of the document title, a field to allow insertion of a filename, and a field to allow the selection of a predetermined document template. General details of the creation of HTML interactive forms are well known.

It is specified by an HTML command starting the form with <FORM> and closing the form with </FORM>.

The title fields may for example be specified by HTML text such as:

```
<INPUT TYPE="text" NAME="document title" SIZE=40 MAX  
LENGTH=100>
```

to set a maximum document title length of 100 characters visible within a screen window of 40 characters.

The document filename field may likewise follow the same format (i.e. the normal HTML format for input of a line of text) but with the substitution of

NAME="document file name" for the string NAME="document title" in the string above.

The document template field may comprise a text area allowing the user to input a document template directly in HTML, in which case it may be specified by an HTML command:

5 <TEXTAREA NAME="TEMPLATE" ROWS=8 COLS=80>
 </TEXTAREA> (which, in this example, creates a field 8 lines high by 80 characters wide).

Alternatively, the template may be provided as a selection from a list of default or prototype templates, for example using the SELECT mechanism provided by HTML; in this case, the list of templates could be represented as
10 the following HTML commands:

 <SELECT NAME="list 1" SIZE=5>
 <OPTION SELECTED> OPTION 1
 <OPTION> OPTION 2

15 </SELECT>

Also provided is a button to allow the user to cause the completed information to be transmitted to the server computer (e.g. using the <SUBMIT> element of the HTML language).

Once the user has filled out the file name and document name and
20 selected a template (or allowed the default template to remain) and has selected the button to submit or upload the form (step 305), the browser program 100 causes the terminal computer 10 to send the contents of the form back to the

server computer 20 at which the server program 110 recognises the contents of the form as such (e.g. by the presence of a "cgi" element within the URL returning the form) and passes the contents of the form to the document management program, which returns them to the document creator.

5 The document creator subprogram 115 reviews the submitted document file name.

 Firstly, the document creator views the file name to see whether it includes portions which cannot validly exist within an HTML document path.

10 If (step 310) invalid characters or strings exist within the file name, then (step 315) the document creator subprogram 115 supplies a failure report message to the server program 110 for transmission to the terminal computer 10.

 If the name is of itself valid, then (step 320) the document creator
15 program 115 accesses the document database 150 to determine whether any existing documents already have the same file name; if so, a failure message is likewise sent (step 315).

 If the file name is acceptable, then (step 325) the document creator
subprogram 115 causes the addition of a new record to the document database
20 store 150, and creates the first version of the new document by editing the document title into the title command within the header element of the new document.

The new document is then written into the document store 155 (step 330), and the document creator subprogram 115 passes a success message to the server program 110 for transmission to the terminal computer 20.

Thus, at the conclusion of the document creator subprogram, the document store includes a new document, comprising a template setting forth the basic document structure elements (e.g. header and sections) and including the new document title, and corresponding records exist in the document database 150. The document URL includes the URL specifying the document management software 102 as well as the document file name.

10 Document editing (125 Figure 1b; step 265 Figure 2)

Referring to Figure 4, the document editing process will now be described in greater detail.

The document editor subprogram accesses the document store 155 to retrieve the document corresponding to the index page from which the editing was specified. Next (step 400) the document editor subprogram 125 calls the document parser subprogram 140, which reviews the document and locates all hypertext labels within the document (corresponding to section headings therein).

Next (step 402) the document editor program 125 edits the document, by inserting an additional hypertext link adjacent each such section heading label. The hypertext link specifies an inline image which, when reproduced,

displays an icon which may be selected by a user to edit the text of the following section.

One further such editing icon is inserted at the beginning of the document.

5 The document thus edited is sent (step 402) to the server program 110 for transmission to the terminal computer 20, at which it is displayed by the browser program 100.

As well as the editing icon inline image, each document editing command inserted specifies the position within the document (e.g. the label)
10 where it is located.

On selection by the user of one of the editing icons, the browser 100 causes the terminal computer 10 to transmit the URL including the specification of the point in the document of the editing icon, which is passed by the server program 110 at the host computer 20 to the document management software
15 102, and then to the document editor subprogram 125.

If (step 412) the icon selected is that at the top of the document, this corresponds to a desire by the user to edit the overall document structure or title.

On receipt of a URL indicating that the document title or structure is to be edited (step 412), the document editor subprogram 125 invokes the lock
20 manager program 145 in a process generally denoted 410 in Figure 4.

Firstly, (step 406) it is determined whether the lock manager subprogram 145 records that the document is already being edited (and is therefore deemed locked), in step 406.

If not, the lock manager subprogram 145 sets a flag in the document
5 database record for the document to indicate that the document is currently locked by the user, in step 416.

If a lock flag is already set (indicating that the document is currently being edited), the lock manager 145 generates a message including details of the other user (including user name, Internet address and date that the document
10 was locked) and passes this to the server program 110 for transmission to the terminal computer 20, which is displayed by the browser program 100.

The message comprises an HTML document including a displayed "override" button which, if selected by the user, causes the transmission of a URL which is passed by the server program 110 to the lock manager 145.

15 If no such message is received (step 408) the editor process terminates. If the user does wish to override the lock (step 408), the lock manager sets a further flag indicating that the document is locked by the current user (step 416).

Thus, the action of the lock manager 145 in locking a document is advisory, leaving it open to the user whether to edit a document which is already
20 being edited by another.

After locking the document (step 416), the document editor subprogram 125 creates a new hypertext document including an editable form portion, and

copies into the editable form (e.g. in a <TEXTAREA>) default text consisting of the current title, followed by a table of contents comprising a list of the section headings within the document (i.e. the text specified at the section heading labels within the document) followed, in each case, by the first few
5 words of the body of the text underneath. Also included is a button to upload or submit the edited form back to the server computer 20, as discussed above.

The editable form is sent (step 418) to the computer 10, and displayed by the browser program 100, and the contents may be edited (step 420) by the user, for example to change the wording of the title or of one of the section
10 headings.

The user may also interchange the order of the section headings, delete a section heading, or add a new section heading within the list.

When the upload button within the form is selected by the user (step 422) the browser program 100 cause the edited form to be returned via the
15 Internet 105 to the server computer 20 (step 424), and the server program 110 forwards the edited form to the document editor subprogram 125.

On receipt, the document editor subprogram 125 causes the version manager subprogram 135 to create a new version number for the document, having a version number one higher than the last stored version number. A
20 corresponding record is created within the document database 150.

Next, the document editor reviews the received form and compares it with the transmitted form; retrieves once more the previous version of the

document from the document store 155; and edits the previous version to produce the text of the new version, as follows:

If a section heading has been deleted in the form, the corresponding section heading and section text below are deleted in the HTML document.

5 If a section heading has been moved, the corresponding section heading and text in the HTML document are moved so that the structure of the edited HTML document reflects the new order of headings in the form.

If the title has been edited or one of the section headings has been edited, the new text of the title or section replaces the old text.

10 If a section heading has been inserted, a corresponding new section heading is created before the following section heading.

If the opening words of text from under one section heading have been moved or copied to lie under another, then the whole text under that section heading within the actual hypertext document is copied to lie under the
15 corresponding section heading.

Thus, radical changes to the structure of the document can be effected by the user simply by manipulating the table of contents.

The edited document is then stored within the document store 155 as the current version.

20 Finally (step 428), the document is unlocked.

If an editing icon within the body of text of the document has been selected (step 414), the document editor 125 again calls the lock manager 145 to

follow the steps denoted as step 410, in which steps 430, 432 and 434 correspond to steps 406, 408 and 416 fully described above.

After execution of the document lock manager, in step 436, the document editor 125 (causes the version manager 135 to retrieve the latest
5 version of the document from the document store 155, and locates the label in the body of the document corresponding to the editing icon selected by the user.

The document editor then modifies this document, by placing form start and end commands (<FORM>;</FORM>) around the section, and specifying the section as an editable text area (<TEXTAREA>) within which the existing
10 text becomes default text. As discussed above, an update, upload or submit button command is also included within the form.

The amended document, including the editable form section, is then passed from the document management software 102 to the server program 110 for transmission to the terminal computer 10, at which the browser program 100
15 displays the document. However, on reaching the editable section, the browser does not act upon the embedded HTML commands, but instead simply displays them as ASCII characters within the editable text area.

The user may then (step 438) edit the text, either to change the text portions, or to change, add or delete the HTML commands embedded within the
20 text.

On completion of editing, the user uploads the form (steps 440 and 442 corresponding to steps 422 and 424 discussed above); and the document editor

and version manager create a new version of the document in step 444 as discussed above in relation to step 426.

Then, also shown within step 444, the document editor 125 retrieves the last version of the document from the document store 155, deletes the section
5 which has just been edited, and substitutes the contents of the editable form for the deleted section, the result being stored as the latest version of the document in the document store 155. Finally, in step 446, the document is unlocked as in step 428 above.

Thus, in this embodiment, the terminal computer only returns the form
10 containing the section of the document which has been edited, which typically represents only a small part of a lengthy document, thereby greatly reducing network traffic.

The means of editing the editable form may, of course, include the use of HTML editors, HTML filters or HTML assistants at the terminal computer
15 10.

Document issuing (120 Figure 1b; step 275 Figure 2)

Referring to Figure 5, the operation of the document issuer subprogram 120 will now be disclosed in greater detail.

When, in step 270 of Figure 2, a user indicates a desire to stabilise the
20 latest version of a document as an issue, on receipt of the corresponding message the document issuer subprogram 120 interrogates the document version manager subprogram 135 to determine whether the current version of a

document is also an issue (i.e. whether no edits have been performed since the last issue).

If so, then in step 510, the document issuer generates a message for transmission back to the user as a hypertext document, which is forwarded to the server program 110 for transmission back to the terminal computer 10, where it is displayed by the web browser 100. The message indicates that the current version has already been issued and that no further action will be taken. The document issuer process then terminates.

If not (step 505) the document version manager reports that the current version of the document has not been issued yet, the document issuer program 125 requests details of all versions of the document since the last issue, which are supplied from the document database 150, and deletes the records of all such documents in the document database 150, and the documents themselves in the document store 155.

It then (step 515) causes the document version manager to renumber the record in the document database to record that the current (i.e. latest) version is to be renumbered as the new issue.

For example, if the current version is the eleventh amended version since the fourth issue, it will previously have been recorded in the document database as version 4.11 (the first numeral indicating the last issue number and the second indicating the version number) form. It will now be renumbered as issue 5 (version 5.0).

Additionally, if the document contains a line beginning <!--IssueStatus-->, the document version manager 135 appends the current issue or version number and the date, and user name of the user who accessed the document. This function may also be performed after editing the document when the
5 version number is changed.

Finally (step 520), the document issuer subprogram 120 supplies to the server program 110 a message for transmission back to the terminal computer 10 indicating that the document has now been issued and giving the issue number, for display via the browser program 100.

10 Thus, after execution of the document issuer, no working versions of the document exist, and the most recent working version has been transformed into the current issue.

Document Deletion (130 Figure 1b; Step 285 Figure 2)

Referring to Figure 6, the operation of the document deleter subprogram
15 130 will now be disclosed in greater detail.

Initially, the document lock manager 145 is invoked in steps 600 and 630 (corresponding to steps 406 and 408 described above).

Instead of performing step 416 described above, if the document is not locked or if the user wishes to override the lock, then (step 605) the document
20 deleter 130 passes a message for transmission to the terminal computer 10 warning that all versions of the document will be deleted. The message document includes a "proceed" hyperlink and a "cancel" hyperlink. When the

user selects one of the hyperlinks (e.g. by clicking a mouse over the corresponding button display) the browser program 100 sends back a URL which is passed to the document deleter 130 from the server program 110 for action.

5 If (step 610) the selected link is determined as the "cancel" hyperlink, the deletion process terminates.

Otherwise (step 610), the document deleter passes the server program 110 a message document listing the versions which are to be deleted (step 615), then instructs the version manager 620 to delete all versions and all issues of the
10 document from the document store 155, and all records of the document and versions from the document database 150 (step 625).

Thus, after operation of the deleter, no trace of the document remains.

Document version manager (135 Figure 1b)

The document version manager functions as described above. In
15 general, it maintains three index page documents which may be accessed by terminal computers; the document list for editing, which (as described above) is available to only those with authoring access rights; a current issue index page listing the current (i.e. highest numbered) issues of all documents in the document store 155 with hypertext links to those documents; and a current
20 version index page, listing current (i.e. highest numbered) versions of those documents (with hypertext links to each). These index pages also list the dates

of creation, and authors, of the corresponding issues/versions, providing an "audit trail" documenting the history of the document.

Document version rollback (147 Figure 1b; step 295 Figure 2)

5 A user may decide that an edited version of a document should be cancelled, and may wish to start working from the previous draft to derive a substitute edited version. This reversion is referred to a "rollback" to the previous edited version; this will be understood to refer to the operation of making the previous version the current version and deleting the most recent edited version.

10 Referring to Figure 7, the operation of the document rollback subprogram 147 will now be disclosed in greater detail.

Initially, the document lock manager subprogram 145 is invoked (steps 700 and 705, corresponding to steps 406 and 408 above) to determine whether the document is locked and, if so, whether the user wishes to override the lock.

15 If the document is not locked, or if the user wishes to override the lock, then (step 710) the document version manager is asked for the user name of the person who created the previous version of the document. Each user is only permitted to cancel edited versions of a document if he himself was the author of the edited version.

20 Thus, if in step 715, it is determined that the two user names are different, then (step 720) the subprogram sends a hypertext document back to the user indicating that the rollback request has been rejected, which is

forwarded to the server program 110 for transmission back to the terminal computer 10, where is it displayed by the web browser 100.

The message indicates that the user cannot undo any changes made by another user and that no further action will be taken. The document rollback
5 process then terminates.

If the current user name is the same as the user name of the user who created the previous version of the document, then (step 725) the document version manager subprogram 135 is invoked to determine whether the latest version of the document is also the latest issue of the document (in other words,
10 as in the document issuer subprogram, to determine whether the document has not been edited since the most recent issue).

If so, then (step 730) the document rollback subprogram generates a message for transmission back to the user as in step 720 above, the message indicating that the current version has already been issued and that no further
15 action will be taken. The subprogram then terminates.

In the event that the current version is not an issue, then (step 735) the subprogram 147 generates a message for transmission back to the user as a hypertext document (as described above in relation to step 720), the message warning the user that the current version of the document will be replaced by the
20 previous version.

The message document includes two hyperlink button icons; one labelled "proceed" and one labelled "cancel". When the user selects one of the

hyperlinks, the browser program 100 sends back a corresponding URL which is passed to the document rollback subprogram 147 from the server program 110.

If (step 740) the selected hypertext link was the "cancel" hypertext link, then the process terminates. If not (step 745), the subprogram passes to the server program 110 a message which indicates that the previous version of the document is no to become the current version. This message is transmitted back to the terminal computer 10 for display by the browser program. The document version manager subprogram 135 is then invoked to remove (i.e. delete) the latest version of the document, and to modify the document database 150 to record that the previous version of the document is once more the current version of the document (step 750). The process then terminates.

Thus, after execution of the process of Figure 7, the most recent edited version of the document is deleted and the previous version will now form the basis for any subsequent edited versions.

15 Other embodiments, variations and modifications

The above described embodiment may, of course, be varied in many respects which will be apparent to the skilled reader.

For example, whilst the invention has been described for use over the Internet, aspects are applicable to other types of computer network. Equally, whilst the invention has been described in relation to hypertext documents, its use for other documents with embedded control characters, or documents in general, or files of data in general (e.g. image or sound files) is not excluded.

Rather than retransmitting the entire document including the section to be edited to the terminal computer, as described above, it would be possible merely to send the section to be edited.

5 Rather than converting portions of a document into an editable form when they are to be edited, it would be possible to store the document in both its original form and with each section already converted. However, this would at least double the storage capacity required, and could increase the possibility of inconsistencies.

10 These, together with any and all other such variants, are intended to be comprehended within the scope of the present invention.

CLAIMS

1. Apparatus for use in a file storage and transfer system, the apparatus comprising a computer with a file store for storing at least a first file,
5 and a communications port for sending said first file via a communications network towards a destination at which it may be edited and for receiving an edited file therefrom, in which the apparatus is arranged to recognise separate portions of said file, and in which during stages of said editing, only portions of said file which are to be edited are carried over said network..
10
2. Apparatus according to claim 1, in which said file comprises a text document.
3. Apparatus according to claim 1, in which said file comprises text
15 with embedded control characters.
4. Apparatus according to any preceding claim, in which said file comprises a hypertext document.
- 20 5. Apparatus according to claim 4, arranged to locate said separate portions and to generate an amended document by inserting at least one editing hyperlink, associated with corresponding portions of the document.

6. Apparatus according to any preceding claim, arranged to supply
an entire said first file to said port, to receive only portions of said first file
which have been edited, and to generate an edited version of said first file
5 therefrom.

7. Apparatus according to claim 4 or claim 5, arranged to recognise
a signal received at said port corresponding to a selection of a link within said
first file associated with a said predetermined portion of said hypertext
10 document, and to generate an editable version of said portion for transmission to
said destination.

8. Apparatus according to any preceding claim, in which said
destination comprises a computer running a hypertext browser program.
15

9. Apparatus according to any preceding claim, comprising a
computer running a hypertext server program.

10. Apparatus according to any preceding claim, further comprising
20 a status store arranged to store status data for the or each said file.

11. Apparatus according to claim 10, in which, on receipt of said edited file, a new version of said first file is created and stored.

12. Apparatus according to claim 7, in which said editable version
5 comprises an editable hypertext form.

13. A document authoring system comprising at least one server computer at which documents are stored and at least one editing terminal to which documents are supplied from said server computer, at which said
10 documents are edited, and from which editing data is returned to said server computer, the server computer being arranged to manage incremental edited versions of said documents.

14. The system of claim 13, comprising:
15 means for receiving an edited version of a said document, for allocating a new version number to said edited version, and for storing said edited version.

15. The system of claim 13, comprising:
20 means for deleting said edited version and for allowing further editing of a previous version of said document.

16. The system of claim 13, comprising:

means for converting a working draft of a said document into an issue of said document.

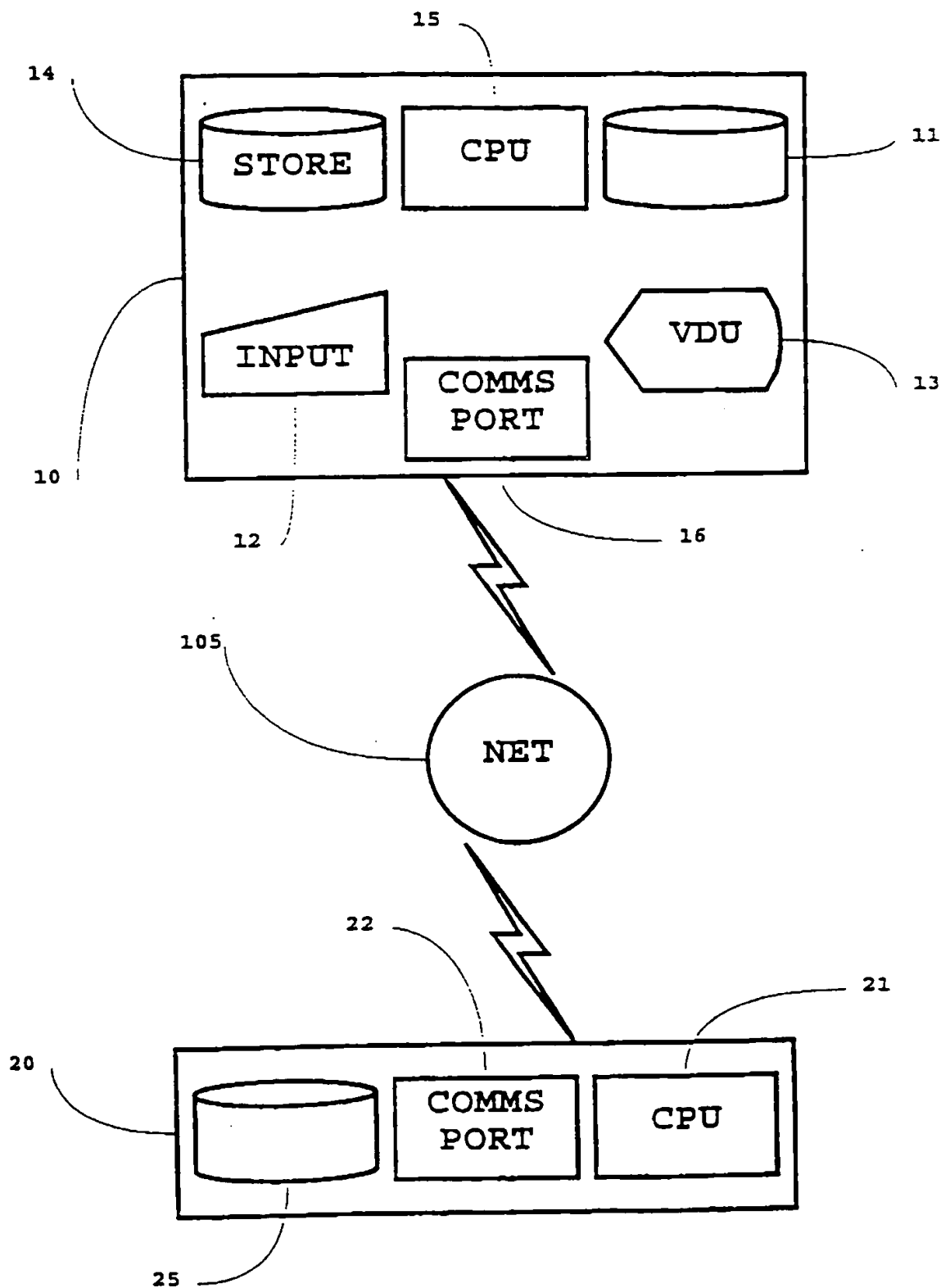
5 17. The system of claim 16, further comprising;

means for deleting previous working drafts of said document.

18. Apparatus for use in a file storage and transfer system, the apparatus comprising a computer with a file store for storing at least a first file,
10 and a communications port for sending said first file via a communications network towards a destination at which it may be edited and for receiving an edited file therefrom, in which the computer is arranged to retrieve the first file from the file store, and to alter said first file to transform it into an editable version, and to send the editable version to the destination.

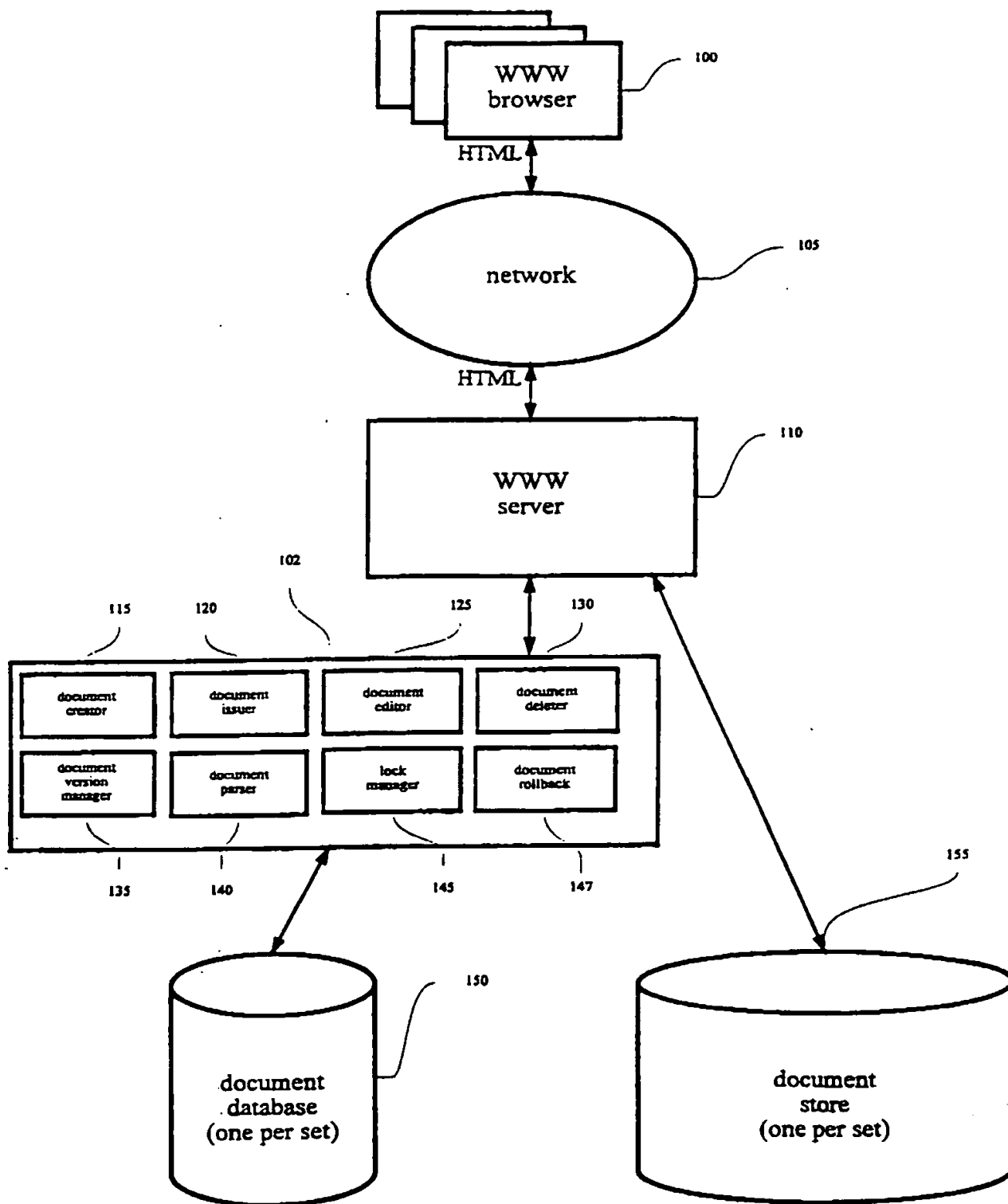
15

1/9

*Fig. 1a*

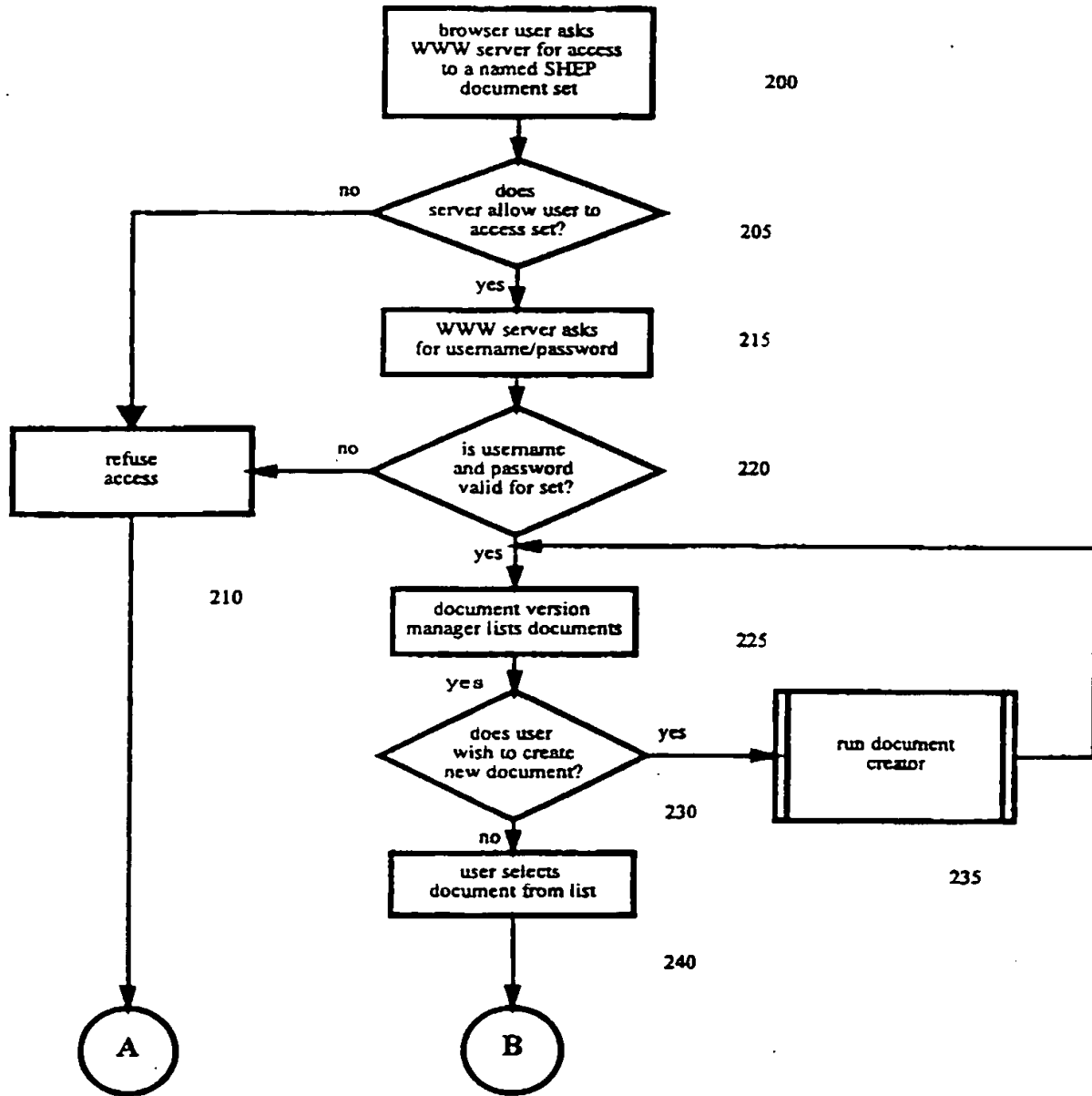
SUBSTITUTE SHEET (RULE 26)

2/9

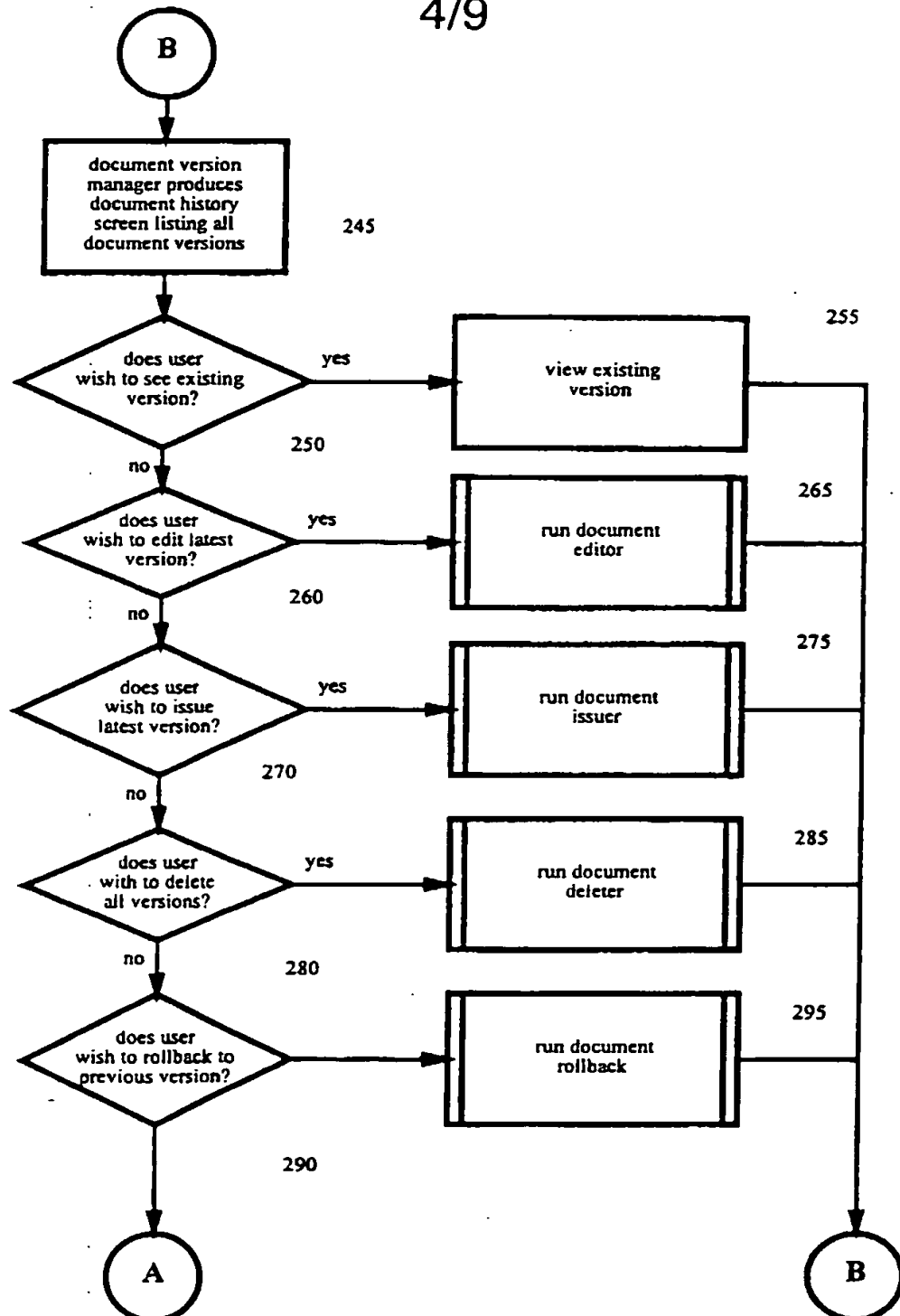
*Fig. 1b.*

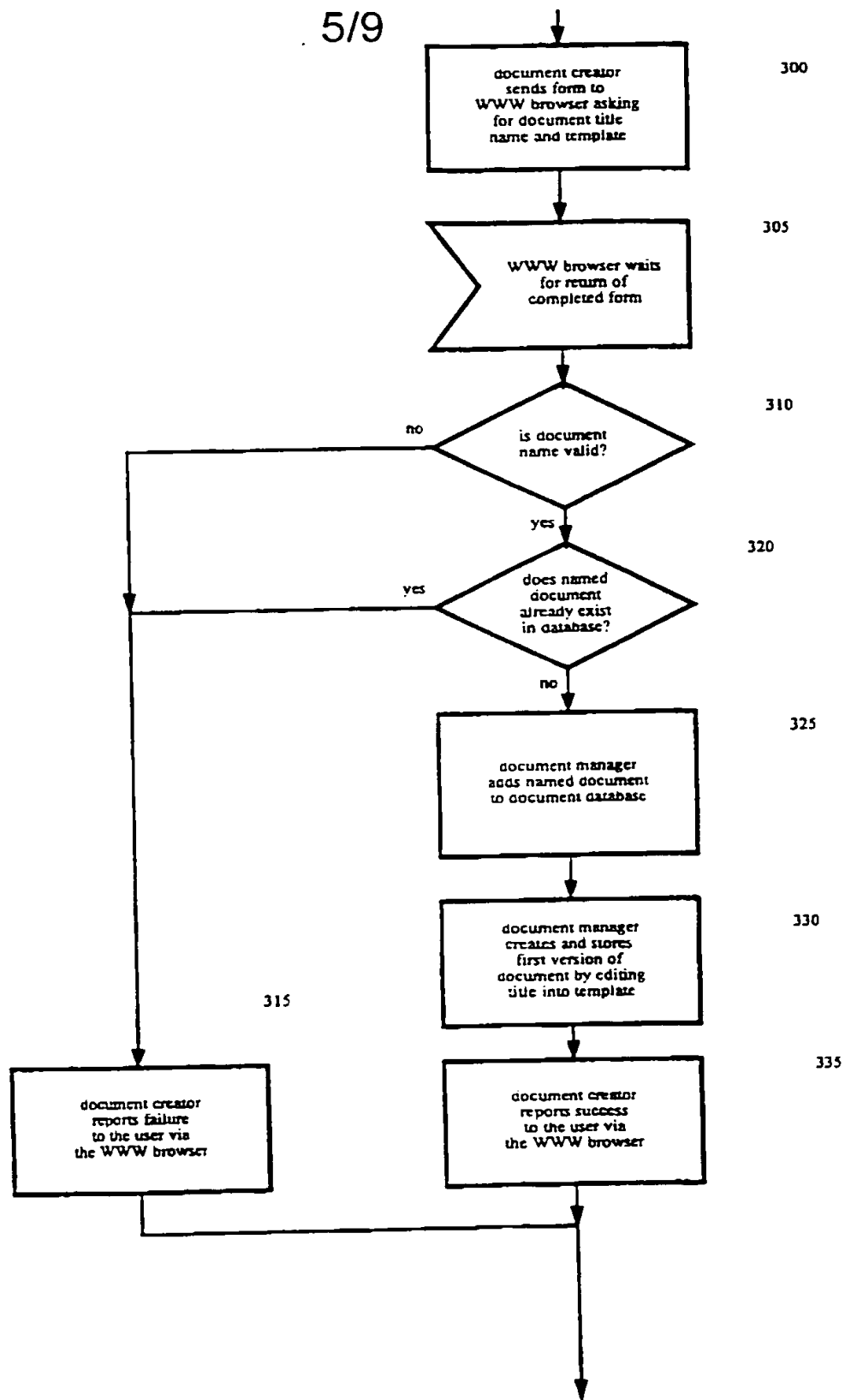
SUBSTITUTE SHEET (RULE 26)

3/9

*Fig. 2a.*

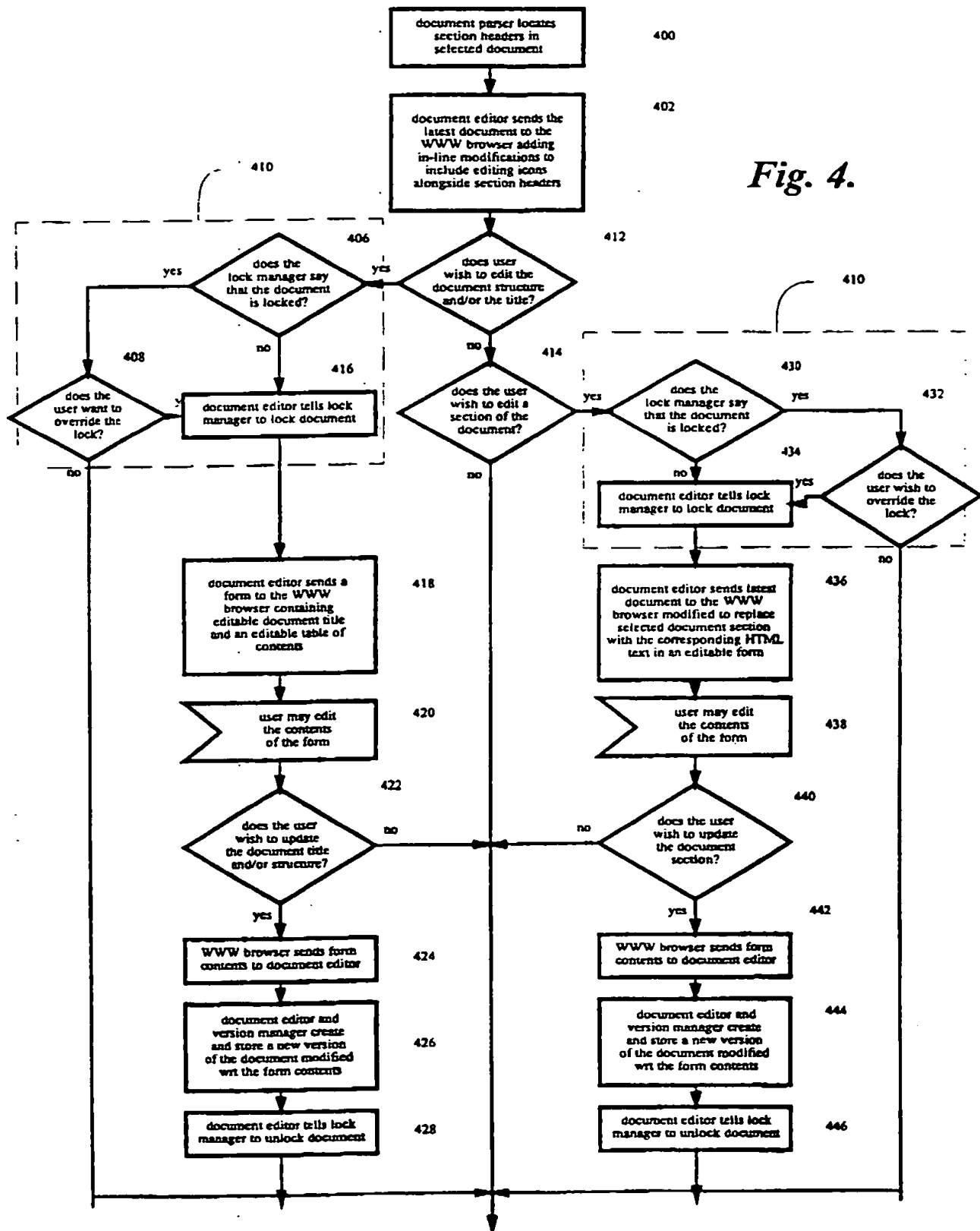
4/9

*Fig. 2b.*

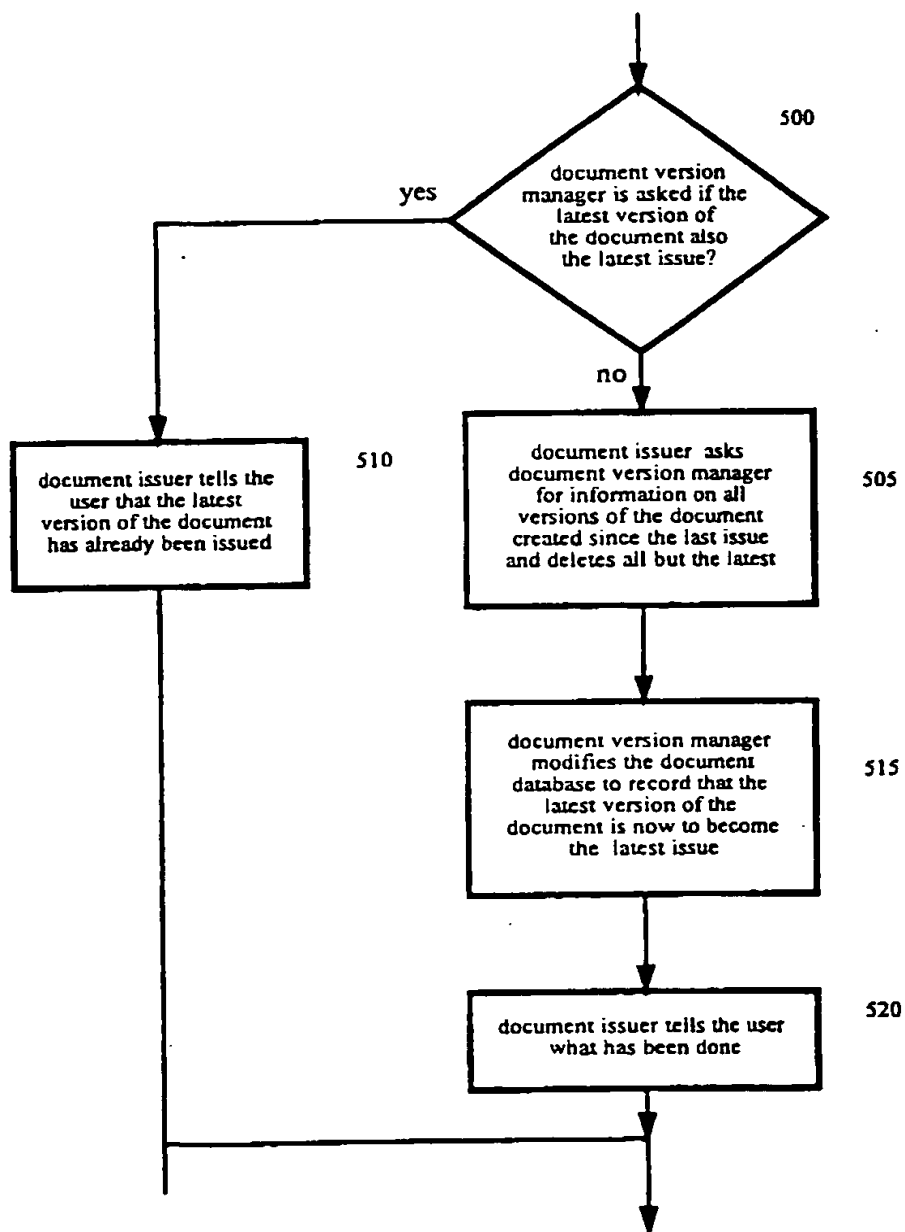
*Fig. 3.*

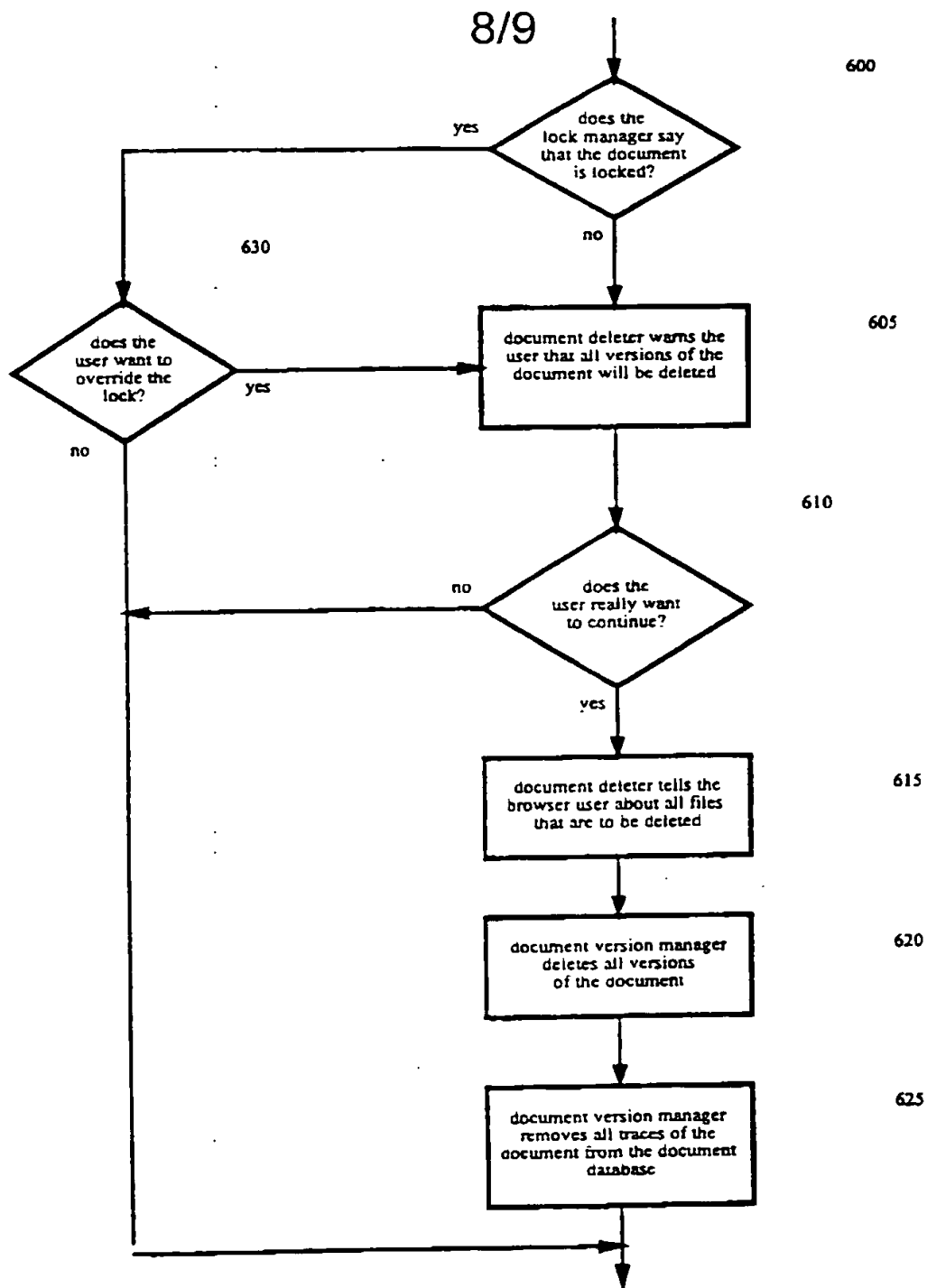
6/9

Fig. 4.



7/9

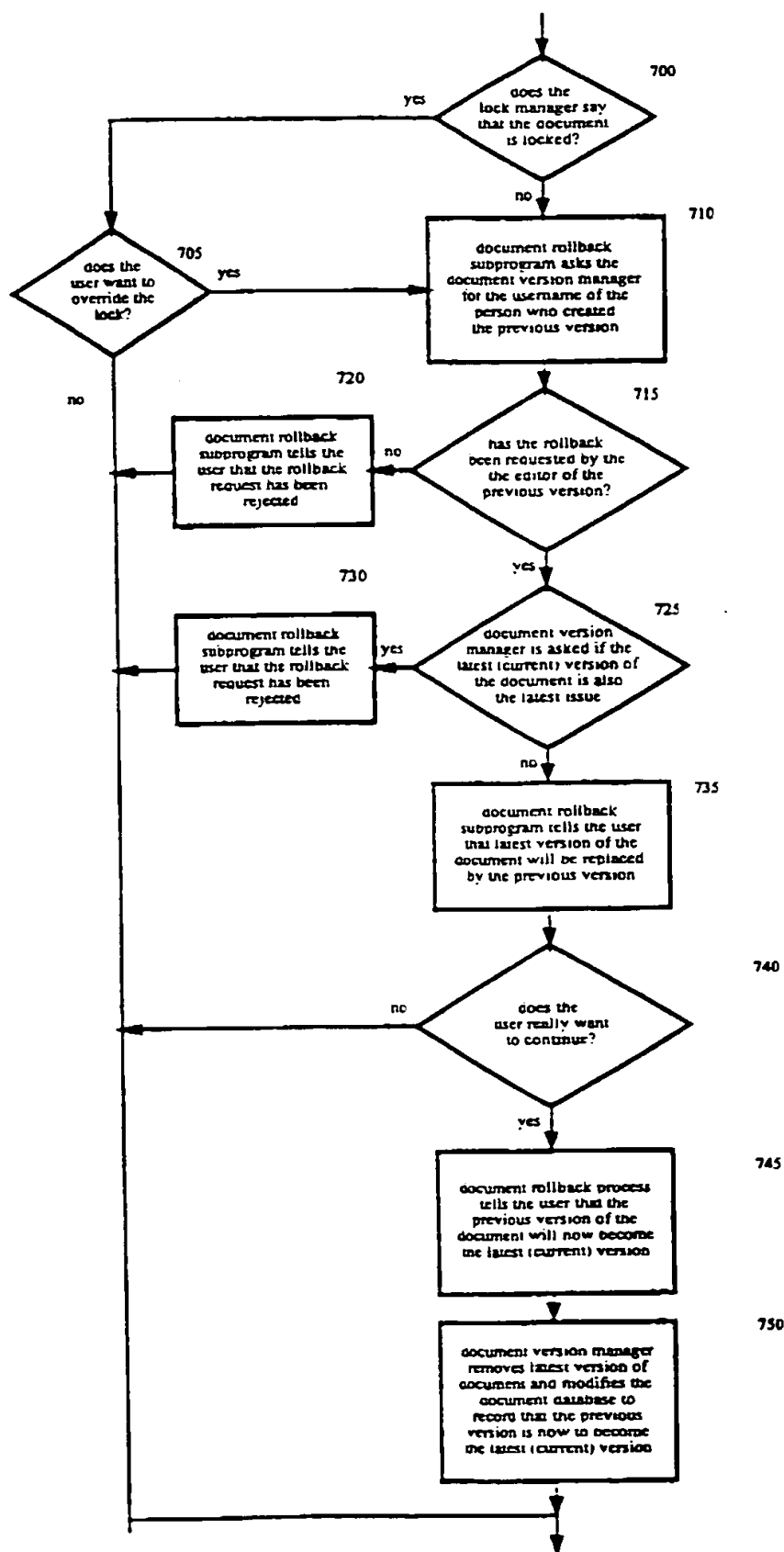
*Fig. 5.*

*Fig. 6.*

SUBSTITUTE SHEET (RULE 26)

9/9

Fig. 7.



INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 97/02887

A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
------------	--	-----------------------

A	WO 95 34032 A (MOTOROLA INC) 14 December 1995 see abstract; claim 1	1-19
---	--	------

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

17 February 1998

Date of mailing of the international search report

26/02/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

Katerbau, R

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 97/02887

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9534032 A	14-12-95	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)